TERMINATION OF NEWTON/CHORD ITERATIONS AND THE METHOD OF LINES *

C. T. KELLEY†, C. T. MILLER‡, AND M. D. TOCCI†

Abstract. Many ordinary differential equation and differential algebraic equation codes terminate the nonlinear iteration for the corrector equation when the difference between successive iterates (the step) is sufficiently small. This termination criterion avoids the expense of evaluating the nonlinear residual at the final iterate. Similarly, Jacobian information is not usually computed at every time step, but only when certain tests indicate that the cost of a new Jacobian is justified by the improved performance in the nonlinear iteration. In this paper, we show how an out-of-date Jacobian coupled with moderate ill-conditioning can lead to premature termination of the corrector iteration and suggest ways in which this situation can be detected and remedied. As an example, we consider the method of lines solution of Richards' equation, which models flow through variably-saturated porous media. When the solution to this problem has a sharp moving front, and the Jacobian is even slightly ill-conditioned, the corrector iteration used in many integrators can terminate prematurely, leading to incorrect results. While this problem can be solved by tightening the tolerances for the solvers used in the temporal integration, it is more efficient to modify the termination criteria of the nonlinear solver and/or recompute the Jacobian more frequently. Of these two, recomputation of the Jacobian is the more important. We propose a criterion based on an estimate of the norm of the time derivative of the Jacobian for recomputation of the Jacobian and a second criterion based on a condition estimate for tightening of the termination criteria of the nonlinear solver.

Key words. termination of nonlinear iterations, Richards' equation, method of lines

AMS subject classifications. 65H10, 65M20, 65N40, 76S05

1. Introduction.

This paper is motivated by some numerical observations made in [15] and [23]. In those papers, we considered a method of lines (MOL) solution of Richards' equation, [20], which is a model of variably-saturated porous media flow. The method of lines uses an ordinary differential equation (ODE) or differential algebraic equation (DAE) solver for temporal integration of a time-dependent partial differential equation that has been discretized in space. Our work to date has focused on backward differentiation formula (BDF) methods that use a predictor-corrector approach. The termination criterion and convergence rate estimator for the corrector iteration in many codes have been designed, at least in part, with the method of lines in mind, and they work well for most problems. For the initial boundary value problem considered in this paper, however, even moderate ill-conditioning coupled with rapid temporal variation of the Jacobian can lead to premature termination of the corrector iteration. In turn, incorrect results may be produced for the entire integration unless one is willing to specify a tight tolerance for the local truncation error. In this paper we give two estimates (2.7) and (2.8) that quantify these effects. We suggest two remedies, recomputation of Jacobians and tightening of the termination criterion for the nonlinear corrector iteration, and criteria for their application. Of

*Version of April 20, 1997. This research was supported by Army Research Office grant #DAALA03-92-G-0111, a Cray Research Corporation Fellowship, National Science Foundation grant #DMS-9321938, US Army contract #DACA39-95-K-0098, and a U. S. Department of Education GAANN fellowship. Computing activity was partially supported by an allocation from the North Carolina Supercomputing Center.

† North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, N. C. 27695-8205, USA (Tim_Kelley@ncsu.edu, mdtocci@unity.ncsu.edu).

‡ Department of Environmental Sciences and Engineering, 104 Rosenau Hall, University of North Carolina, Chapel Hill, NC 27599-7400 (casey_miller@unc.edu).

the two, more frequent recomputation of the approximation to the Jacobian of the corrector equation J, which we apply based on either an estimate of $\|dJ/dt\|/\|J\|$ or consideration of the Newton step, has the most significant effect. Tightening of the termination criterion for the nonlinear solver, the criterion which is an estimate of the condition number, has a less significant, but still noticeable, effect.

While our numerical results, motivation, and suggested modifications in the frequency of Jacobian evaluation and termination of the corrector iteration are related to a particular problem, the analysis in § 2 is problem independent and the modifications may be more broadly applicable as well.

The application of concern in this work is Richards' equation (RE), which is used to describe variably-saturated water flow in a rigid porous media. RE results from a mass conservation law for a two-fluid system (water and gas) in which the assumption of constant gas-phase pressure has been applied. This assumption is justified for many systems because a very small gas-phase pressure gradient is needed to support the flow of a gas phase compared to the pressure gradient needed to support an equal volumetric flow of an aqueous phase. Constitutive relations are required to close the conservation law; we detail this formulation below.

The pressure head form of RE in one space dimension is [6]

$$[c(\psi) + S_s S_a(\psi)]\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial z}\left[K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\right] \tag{1.1}$$

where ψ is pressure head; $c(\psi) = \partial\theta/\partial\psi$ is the specific moisture capacity; $\theta(\psi)$ is the volumetric fraction of the water phase; S_s is the specific storage, which accounts for the slight compressibility of water; $S_a(\psi) = \theta(\psi)/n$ is the aqueous-phase saturation; n is the porosity of the porous media; and $K(\psi)$ is the variably-saturated hydraulic conductivity. In this formulation, the z axis is the vertical direction oriented positively upward. In order to close (1.1), relations among ψ, $\theta(\psi)$, and $K(\psi)$ must be specified. The relations used in this work are the van Genuchten [24] and Mualem [16] relationships. First, we define the effective saturation, S_e, using the van Genuchten relation:

$$S_e(\psi) = \frac{\theta - \theta_r}{\theta_s - \theta_r} = (1 + |\alpha_\nu \psi|^{n_\nu})^{-m_\nu} \tag{1.2}$$

where θ_r is the residual volumetric water content, θ_s is the saturated volumetric water content, α_ν is an experimentally-determined coefficient that is related to the mean pore size, n_ν is an experimentally-determined coefficient related to the variation in pore sizes, and $m_\nu = 1 - 1/n_\nu$.

The variably-saturated hydraulic conductivity is defined using Mualem's model:

$$K(\psi) = K_s S_e^{1/2}\left[1 - \left(1 - S_e^{1/m_\nu}\right)^{m_\nu}\right]^2 \tag{1.3}$$

where K_s is the water-saturated hydraulic conductivity. The parameters in the conductivity and saturation influence the speed and slope of moving fronts in the solution and thereby make the problem more or less difficult.

In [15] and [23], we used the DAE code DASPK, [5], [2], [18], with the direct solver option, for temporal integration of RE in one space dimension. In these papers, we showed experimentally that if certain modifications were made to the nonlinear iteration for the corrector phase of the integration, then the MOL was both more efficient and more robust than several alternative methods commonly used in the

hydrology community. The purpose of this paper is to analyze those modifications in more detail and provide guidance on how they can be adapted automatically to the problem.

Our approach to the MOL in [15] and [23] treated (1.1) as a DAE because the coefficient $c(\psi) + S_s S_a(\psi)$ that multiplies the time derivative can become very small, making the Jacobian ill-conditioned in many backward difference implicit time integration schemes for ordinary differential equations. We examined two porous media types (problems A and B in § 3). We found that the MOL performed best if the termination criteria in DASPK was tightened and Jacobians reevaluated at each time step. Our numerical results here, based on more detailed experimentation, show that only the more difficult of the two problems benefits from a tightened termination criterion for the nonlinear iteration and that, for the easier of the two problems, the Jacobian need not be reevaluated at every time step. Of course, equally accurate results can be obtained by tightening the user-supplied relative and absolute local truncation errors, but solving the problems in this way reduced the efficiency of the MOL solution [23].

2. The Corrector Equation. We will suppress the spatial variable z and let $u(t)$ be the continuous solution at time t. Similarly, u_n will denote the spatially discretized solution at time step n, which will correspond to time t_n. Given the solution u_n, DASPK and many other ODE and DAE codes form a nonlinear equation for the solution at the next time step $t_{n+1} = t_n + \delta_n$. This corrector equation is solved by a variation of Newton's method with a predictor formula used to provide the initial iterate.

We write the corrector equation as

$$F(u_{n+1}) = 0. \tag{2.1}$$

DASPK and similar codes solve (2.1) with the modified Newton or chord method [11]. F may depend on values of the solution at earlier time steps, but we suppress that dependence here. We will denote the iterates for the solution of (2.1) by x_k, to distinguish them from the solution of the PDE of interest u.

Having formed a predictor x_0 (usually not u_n) to the solution u_{n+1} of (2.1) and an approximate Jacobian $A \approx F'(x)$, we attempt to compute $x^* = u_{n+1}$ with the chord iteration

$$x_{k+1} = x_k - A^{-1} F(x_k).$$

2.1. Termination of the Corrector Iteration. Nonlinear iterations in the DAE and ODE integrators mentioned above terminate based upon convergence estimates that are a function of the chord step-length taken

$$s_k = -A^{-1} F(x_k) \tag{2.2}$$

as estimates of the errors $e_k = x_k - x^*$ for $k \geq 0$. If direct methods are used to solve linear systems, it is efficient to use a factorization of an approximate Jacobian A for as many nonlinear iterations and values of t as possible so as to avoid matrix factorizations and evaluations.

Typically, A is the Jacobian of F for some previous iterate but not necessarily at the current time step, and when A is recomputed, the Jacobian for the current values of x and t is used. In this case, if the Jacobians vary rapidly in t then one should recompute Jacobians whenever t is changed. If this action is not taken, q-linear convergence may not take place at all or may be very slow. DASPK and related codes

decide whether to reevaluate the Jacobian by examining the norms of the chord steps. If these norms decrease slowly or not at all, then δ_n is reduced and/or the Jacobian is recomputed. We refer the reader to [2] for more detail. However, [23], an early step in the chord method with an inaccurate and/or ill-conditioned Jacobian can be much smaller than the actual error. In this case, both the test for successful termination of the nonlinear iteration and the error test for local truncation can be incorrectly passed. A partial remedy [23] is to recompute the Jacobian at each time step.

2.2. Convergence Rates. In this section, we review the convergence rate estimates for the chord method for solution of a nonlinear equation $F(x) = 0$. The dependence on the relative error in the Jacobian, the normalized Lipschitz constant for the Jacobian, and the error in the initial iterate are clearly exposed.

If the standard assumptions [11] (Lipschitz continuity and nonsingularity of F' near x^*, a root of F) for local quadratic convergence of Newton's method hold, and if x_0 and A are sufficiently good approximations to x^* and $F'(x^*)$ then the chord method is q-linearly convergent, [11], [8], [17]. This means that there is $\rho \in (0,1)$ such that

$$\|e_{k+1}\| \le \rho \|e_k\| \tag{2.3}$$

for all $k \ge 0$. In (2.3) $e = x - x^*$, and ρ is called the q-factor. It is known, [22], that *a priori* knowledge of the q-factor leads to an effective termination criterion that does not require a costly evaluation of F at the terminating iterate. However, estimates of the q-factor that depend on the norms of the chord steps can lead to premature termination in the problems we consider here.

We will briefly review the analysis that leads to (2.3) with a view toward quantifying how the conditioning of A and F', the Lipschitz constant γ of F', and the quality of the predictor influence the size of the q-factor. As a byproduct of this analysis, we will see how reliable the step is as an indicator of the size of the error. Our expression of this idea in (2.8) differs from the standard results in [8], [11], and [17] because of its emphasis on the relative error and condition number of the approximate Jacobian rather than on absolute error estimates, which are all that are needed for a conventional convergence proof.

The notation is standard. We set $E = A - F'(x^*)$. We will let x_c denote a current approximation to x^*, $x_+ = x_c - A^{-1}F(x_c)$, the subsequent chord iteration, and $s = -A^{-1}F(x_c)$ the chord step. We begin with the simple observation that $s = x_+ - x_c = e_+ - e_c$, and so

$$\frac{\|s + e_c\|}{\|e_c\|} = \frac{\|e_+\|}{\|e_c\|}. \tag{2.4}$$

Hence the relative accuracy of approximating $\|e_c\|$ by $\|s\|$ is roughly equal to the q-factor. We can conclude, as in [21], that termination on small steps is an effective approach only if the convergence is fast.

Now if A is nonsingular, and x_c is near enough to x^*, then

$$\begin{aligned} e_+ &= e_c - A^{-1}\int_0^1 F'(x^* + te_c)e_c\,dt = A^{-1}\int_0^1 (A - F'(x^* + te_c))\,e_c\,dt \\ &= A^{-1}(A - F'(x^*))e_c + A^{-1}\int_0^1 (F'(x^*) - F'(x^* + te_c))\,e_c\,dt \end{aligned} \tag{2.5}$$

and

$$\|e_+\| \le \|A^{-1}\|(\|E\| + \gamma\|e_c\|/2)\|e_c\|. \tag{2.6}$$

The estimate (2.6) is the one usually used to prove local convergence of Newton's method and the chord method. Now, if $e_c \neq 0$,

$$\frac{\|e_+\|}{\|e_c\|} \le \kappa(A)\left(\frac{\|E\|}{\|A\|} + \frac{\gamma\|e_c\|}{2\|A\|}\right). \tag{2.7}$$

Therefore, using (2.4) and (2.7),

$$\frac{|(\|s\| - \|e_c\|)|}{\|e_c\|} \le \frac{\|s + e_c\|}{\|e_c\|} = \frac{\|e_+\|}{\|e_c\|} \quad \le \kappa(A)\left(\frac{\|E\|}{\|A\|} + \frac{\gamma\|e_c\|}{2\|A\|}\right). \tag{2.8}$$

Equation (2.7) estimates the q-factor in terms of the condition number $\kappa(A)$ of A, the relative error $\|E\|/\|A\|$ in A, and the product of the normalized Lipschitz constant $\gamma/\|A\|$ with the norm of the current iterate.

Similarly (2.8) relates these quantities to the relative accuracy of $\|s\|$ as an approximation to $\|e\|$ and shows that even a moderately ill-conditioned A, when coupled with an out-of-date Jacobian (large $\|E\|$), can imply that s is a poor approximation to the error and that convergence is slow even when e itself is small. Note that if only one of the two problems, moderate ill-conditioning or an inaccurate Jacobian, takes place, the estimate (2.7) can still imply that convergence is fast and therefore s is a good approximation to e.

2.3. Termination. If one has *a priori* knowledge of ρ, or at least an upper bound, then (2.3) implies that

$$(1 - \rho)\|e_k\| \le \|e_k\| - \|e_{k+1}\| \le \|s_k\|, \tag{2.9}$$

for $k \ge 0$. Hence $\|e_{k+1}\| \le \rho\|s_k\|/(1 - \rho)$. This inequality was used in [22] and in many papers thereafter to argue that one could terminate the iteration when

$$\frac{\rho}{1 - \rho}\|s_k\| \le \ \sigma\epsilon, \tag{2.10}$$

and conclude that $\|e_{k+1}\| \le \epsilon$ without needing to evaluate $\|F(x_{k+1})\|$. The quantity $\sigma \in (0, 1)$ in (2.10) is a safety factor (called the *discount* in [22]), and is a guard against the possibility that the estimate for the q-factor is too low. The value $\sigma = 0.33$ is used in many codes [3], [5], [18].

2.4. Estimation of the q-factor. In the context of ODE and DAE integration, where one expects to perform only a small number of expensive nonlinear corrector iterations, the savings of a single function evaluation that results from termination on small steps, rather than small residuals, is very valuable. It is important, therefore, that the estimate of ρ be accurate and crucial that it not be a gross underestimate.

Typical estimates of ρ include [4], [18]

$$\rho_k^P = \frac{\|s_k\|}{\|s_{k-1}\|} \text{ and } \rho_k^L = \left(\frac{\|s_k\|}{\|s_0\|}\right)^{1/k}. \tag{2.11}$$

In some codes [18], the estimate for ρ may be carried over from previous values of t, a valid decision if the Jacobian varies slowly as function of t. In the work reported

in [23], the Jacobian varied rapidly as a function of t, and we found it advantageous to recompute Jacobians at each time step in order to accurately estimate the q-factor and terminate the iteration correctly. If $\|s_k\| \approx \|e_k\|$, then either estimate in (2.11) will provide an acceptable estimate. The danger here is that if $\kappa(A)$ and $\|E\|$ are both large, s may be a poor approximation to e and, therefore, the q-factor estimates in (2.11) will be poor as well. Hence, these q-factor estimates should not be used to test the accuracy of s as an approximation to e.

2.5. Problems with the Termination Criteria. From (2.8) we see that if both $\kappa(A)\|E\|/\|A\|$ and $\|e\|$ are not sufficiently small, then it is possible that $\|s\|$ can be small enough to terminate the nonlinear iteration while $\|e\|$ is still unacceptably large. Moreover, the tests for local truncation error may not detect this failure, since both the estimate for ρ and $\|x_0 - x_k\|$ could be small enough to satisfy the tests for a good predictor and small local truncation error. This situation seems to apply to the problems discussed in [15] and [23].

In [23], we found that best performance could be obtained by

- reevaluating (and therefore refactoring) Jacobians at each time step as a response to the rapid variation in the Jacobian and
- reducing the discount σ as a response to ill-conditioning in the Jacobian

In § 3, we consider two RE problems with different material properties. The strategy used in [23] of reducing the discount and reevaluating the Jacobian at each time step is unnecessary for some problems but very appropriate for others. In view of this observation, we advocate a scheme that monitors approximations to the temporal derivative and the condition number of the Jacobian and makes reevaluation decisions and adjusts the discount based on this information.

3. Adaptive Scheme and Numerical Results. In this section, we present numerical results for two different test problems. Each of these problems models a different porous media type and represents varying degrees of numerical difficulty. The physical parameters used in the constitutive laws given in § 1 for K, c, and S and the boundary and initial conditions used for these two problems are presented in Table 3.1. Problem A is the easier of the two with a less sharp and more slowly moving front.

We will denote the Jacobian of the corrector equation by J and the approximate Jacobian by A. In DASPK, A is also the Jacobian for the corrector equation, but perhaps from a previous time step.

We used the standard finite difference discretization in space that was employed in [23] and the direct solver mode of DASPK for temporal integration. The relative, *rtol*, and absolute, *atol*, local truncation error tolerances in DASPK were set to the same value, which we will denote as *tol*, for these experiments. The spatial domain was the interval $0 \le z \le 10$ with $z = 10$ being the surface. The time interval and boundary/initial conditions for each test problem are given in Table 3.1. In the Table, ψ_0 is the initial condition for the pressure head, and $\psi_1 = \psi(z = 0, t)$ and $\psi_2 = \psi(z = 10, t)$ are the left and right Dirichlet boundary conditions.

From (2.7) and (2.8) we see that $\kappa(A)$ and $\|E\|$ can have an effect both on convergence rates and on the accuracy of $\|s\|$ as an approximation to $\|e\|$. When $\kappa(A)$ and $\|E\|$ become even moderately large, it may be necessary to make changes to the nonlinear solver to maintain the accuracy of the solution. We propose a scheme whereby the discount factor, σ, is decreased when $\kappa(A)$ is sufficiently large. We also give two criteria for updating the Jacobian, one based on an estimate of $\|dJ/dt\|$, indicating a likelihood that $\|E\|$ is not sufficiently small, and the other based on how, after a new

TABLE 3.1
Model parameter values for Problem A and Problem B.

Variable	Problem A	Problem B
θ_r	1.02×10^{-1}	9.30×10^{-2}
θ_s	3.68×10^{-1}	3.01×10^{-1}
α_ν (m^{-1})	3.35×10^{0}	5.47×10^{0}
n_ν	2.00×10^{0}	4.26×10^{0}
K_s (m/day)	7.97×10^{0}	5.04×10^{0}
S_s (m^{-1})	0.00×10^{0}	1.00×10^{-6}
z (m)	[0, 10]	[0, 10]
t (days)	[0, 10]	[0, 0.2]
Δz (m)	$0.0125m$	$0.0125m$
Init Δt (days)	1.16×10^{-8}	1.00×10^{-7}
ψ_0 (m)	-1.00×10^{1}	$-z$
ψ_1 (m)	-1.00×10^{1}	0.00×10^{0}
ψ_2 (m)	-7.50×10^{-1}	1.00×10^{-1}

Jacobian has been evaluated and factored, the computed Newton step is affected by that change. The amount of reduction in σ and the specific tests and thresholds for changing σ and recomputing the Jacobian will be specified later.

Before implementing a scheme to automatically handle the changes in the nonlinear solver, numerical experiments on the two test problems were done to examine the effect of reducing σ by a constant factor and recalculating Jacobians at every time step. The solution for Problem B is shown in Figure 3.1. The plots are similar for the other test problem. In the figure, the upper left plot is of the solution (pressure head) as a function of space in the region near the moving front. The steeper plots correspond to the finer meshes. The meshes were $h = 1/80, \ldots, 1/2560$ and the tolerances in DASPK were set to $tol = 1.d - 8$. We will refer to these solutions as the "tight tolerance" solutions. In the other plots larger tolerances were used in an attempt to reduce the cost of the computation. The tight tolerance solutions with $h = 1/80$ and $h = 1/160$ (solid lines) are used to compare to loose tolerance solutions with a spatial mesh of $h = 1/80$. The dashed lines correspond to solutions with $tol = 1.d - 3$ (left) and $tol = 2.d - 3$ (right).

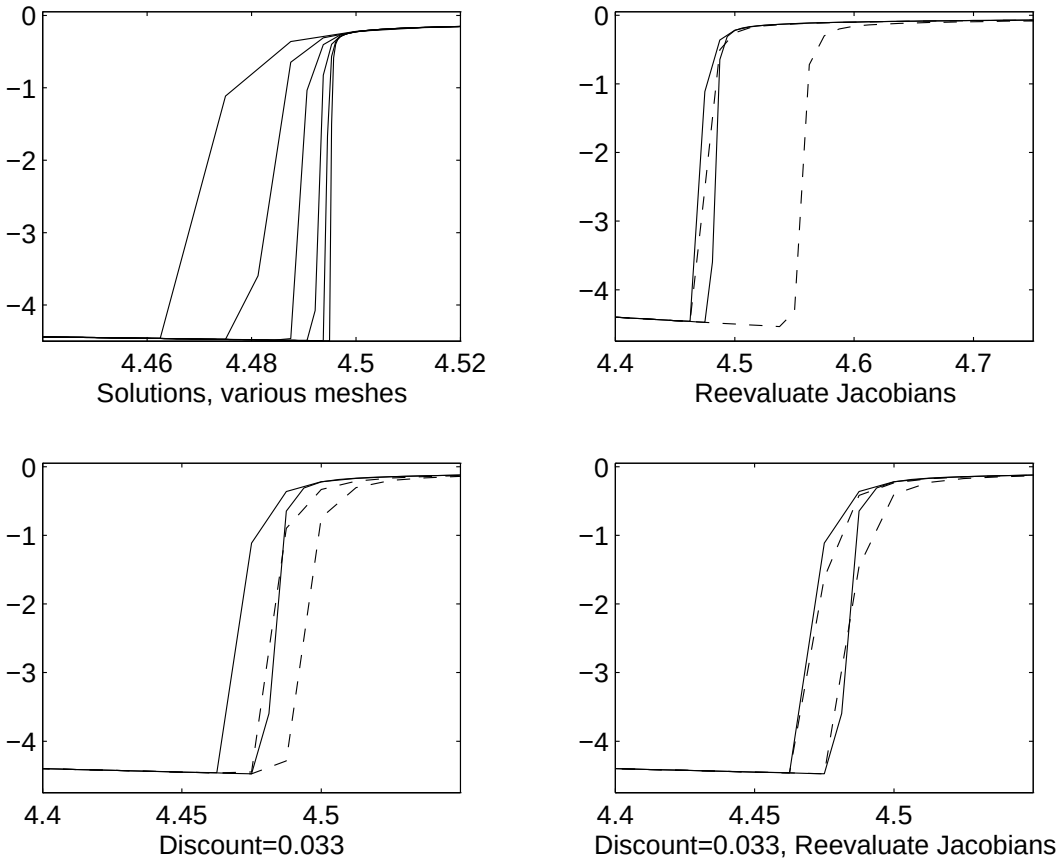
In the upper right plot, Jacobians are updated with every time step, and the standard discount $\sigma = 0.33$ is used; in the lower left plot $\sigma = 0.033$, the value used in [23], and the default Jacobian update method is used; in the lower right plot, both modifications are used. One can see that both modifications together lead to results that are nearer the tight tolerance solution but not necessarily nearer the solution with the finer spatial mesh.

Since not all problems are as difficult as problem B, it is desirable to have an adaptive scheme that detects when it is necessary to make the changes outlined above. A direct approach measures the change in the Jacobian matrix itself. To do this we used the estimate

$$\frac{\|dJ/dt\|}{\|J\|} \approx \frac{\sup_{ij} |(A^{(new)})_{ij} - (A^{(old)})_{ij}|}{\sup_{ij} |(A^{(new)})_{ij}| (t_{new} - t_{old})} = dA \tag{3.1}$$

in order to detect the possibility of large $\|E\|$ (in the relative sense) and the need to reevaluate the Jacobian. In (3.1), $A^{(new)}$ is the current Jacobian computed at

FIG. 3.1. *Dune Sand Problem (problem B)*



current time t_{new}, and $A^{(old)}$ is the previous Jacobian computed at time t_{old}. If dA is sufficiently large, we compute a new Jacobian at the next time point. For our tests, we recompute Jacobians when

$$dA > 0.73. \tag{3.2}$$

We will call this approach matrix-based updating. Later in this section we will report on some experiments with some small DAE test problems from [13]. Those results indicate that the matrix-based approach can become too conservative when large changes in the Jacobian do not result in correspondingly large changes in the Newton step.

In order to determine when a rapid change in the Jacobian would affect the corrector iteration in a significant way, we measure the effect of that change on the computed Newton step. To do this we keep a factored form of $A^{(old)}$ in memory, and when a new Jacobian, $A^{(new)}$, is factored, the first Newton step is computed using $A^{(old)}$ and $A^{(new)}$. If we denote the steps by $s^{(old)}$ and $s^{(new)}$ respectively, then we define,

$$dA = \|s^{(old)} - s^{(new)}\|_{WRMS} \tag{3.3}$$

where $\|\cdot\|_{WRMS}$ is the weighted (in terms of the relative and absolute error tolerances

input to the code) root mean square norm in DASSL. In view of the weighting, we can assume that if $dA > 1.0$ then we must update the Jacobian on the next step. A more conservative approach, which we found to be more effective in our testing, is to update the Jacobian on the next step if

$$dA > 0.1. \tag{3.4}$$

We call this method step-based updating and, in view of our RE results and results on a suite of test problems [13], recommend it over matrix-based updating for general use. However, matrix-based updating was somewhat better for the RE problems. It should be noted that either of these updating methods are intended to be added to the existing Jacobian update strategy in DASPK and are not replacements.

For the problems in one space dimension that we consider in this work, $\kappa(A)$ was found using the LAPACK condition estimator [1]. For problems in more space dimensions, one could use methods such as those discussed in [7], [12], or [14]. Based upon numerical experimentation, we reduce the discount when $\kappa(A) > 10^5$ using

$$\sigma = \begin{cases} 0.33 & \text{if } \kappa(A) < 10^5 \\ 0.33/(1 + 2(\ln \kappa(A) - \ln 10^5)) & \text{if } \kappa(A) \geq 10^5 \end{cases} \tag{3.5}$$

The condition number threshold is not particularly large but is large enough to affect the convergence rate and relative error estimates in (2.5) and (2.8).

When we combine the adjustment in the discount with either the matrix-based or step-based updating scheme for the Jacobian we get two different adaptive methods, which we refer to as AdaptM when matrix-based Jacobian updating is used and AdaptS when step-based updating is used. In Table 3.2 we report on a comparison of the two adaptive methods, the default DASPK scheme, and the best combination of recomputing Jacobians at every step and/or reducing the discount. This best, or tuned, estimate was based upon extensive numerical experimentation, but does necessarily represent optimal values.

The Jacobian in DASPK will change in a benign way if the order or step size is changed by the integrator. This change will be small and should not, by itself, activate either step-based or matrix-based updating. We did test this by deactivating our Jacobian updating methods after a step size or order change. Our experiments indicate that step size and order changes had little effect on our decisions to update the Jacobian.

TABLE 3.2
Results of the adaptive scheme for the two test problems.

		tol	Ave σ	% Jacs Recalc	Func Calls	Jac Calls	Coarse Error	Dense Error
A	Def	2.0e-4	0.33	27.2	4065	526	6.10e-3	9.37e-2
	Tuned	4.0e-4	0.33	100.0	2681	1343	1.86e-3	8.94e-2
	AdaptM	3.0e-4	0.33	56.0	3311	905	6.02e-3	9.36e-2
	AdaptS	3.0e-4	0.33	69.3	2979	1041	1.92e-2	1.07e-1
B	Def	8.0e-5	0.33	36.5	33698	5419	9.21e-3	8.62e-2
	Tuned	1.0e-3	0.033	100.0	14437	6358	5.04e-3	8.97e-2
	AdaptM	8.0e-4	0.054	100.0	14136	6289	7.97e-3	8.71e-2
	AdaptS	2.0e-4	0.058	91.8	18041	7712	1.48e-2	8.13e-2

In Table 3.2, the tolerance was chosen so that the method was accurate and efficient, meaning that the numerical solution had reached spatial truncation error

and further reduction of the tolerance would not improve the quality of the solution. The average σ is just an arithmetic average of the discount factor σ throughout the iteration, where the default value is 0.33. The next column shows the percentage of the times possible where the Jacobian is updated. The next two columns show how many function and Jacobian calls are executed, and the last two columns are error measures. The coarse error is defined by

$$\frac{1}{\Delta z} \sum_{i=0}^{n_n-1} |u_i - u_i^{\mathrm{c}}|$$

where n_n is the number of nodes in the spatial discretization, u^{c} is the solution on the same mesh as u_i with *tol*= 1.0e-8, and the dense error is defined by

$$\frac{1}{\Delta z} \sum_{i=0}^{n_n-1} |u_i - u_i^{\mathrm{d}}|$$

where u^{d} is the injection of the tight tolerance solution with a spatial mesh width of 1/2560 onto the mesh with $\Delta z = 1/80$.

The results from Table 3.2 indicate that the default strategy for the corrector equations requires that the tolerances be set more tightly (with a significant cost in function/Jacobian evaluations) than either hand-tuning or using the adaptive strategies proposed here. For the harder of the two problems, we found that it was better to reduce the discount factor σ. For both of the problems, it is better to recalculate Jacobians more often than the default strategy does.

Although the adaptive schemes described in this paper were designed solely to improve the results for RE, we also implemented the schemes on a set of ODEand DAE test problems, [13]. To be consistent with the test set results we used the code DASSL, rather than the direct solver mode of DASPK, to solve the problems. The test set contains ODE's and index one and higher DAE's. We compared DASSL with the default discount and Jacobian updating rule with AdaptM and AdaptS for the ODE and index one DAE test problems. As expected, for most of the test problems there was very little difference between the default method and either of the adaptive methods. However we saw significant differences in two of the problems. For test problem #4, a system of equations for a ring modulator [10], AdaptM updated Jacobians far too frequently and was much less efficient than the default. AdaptS was much less aggressive in updating the Jacobian and, while less efficient, was within 10% of the default method.

For test problem # 6, a model of a transistor amplifier from [9] and [19], all three methods (default, AdaptS, AdaptM) showed inconsistency in the relation of work and accuracy to the input tolerances in that a reduction in *rtol* and *atol* did not imply either a more accurate or more costly integration. For this reason, we considered a variety of tolerances and looked at averages. With AdaptS we saw a approximately a 30 % average decrease in work for the step-based adaptive scheme as opposed to the default scheme, both methods giving similar accuracy. With AdaptM we saw a 10% in accuracy and a 50 % decrease in work.

The results on the DAE test problems lead us to prefer for AdaptS, which was far superior to AdaptM on problem #4 in the test suite and close to it in performance the two RE test problems and on problem #6 in the test suite.

Acknowledgments. The authors wish to thank Peter Brown, Steve Campbell, and Linda Petzold for several illuminating conversations about DAEs, ODEs, and the method of lines.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users Guide, Second Edition*, SIAM, Philadelphia, 1992.

[2] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, no. 14 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.

[3] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh, *VODE: A variable coefficient ode solver*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1038–1051.

[4] P. N. Brown and A. C. Hindmarsh, *Reduced storage matrix methods in stiff ODE systems*, J. Appl. Math. Comp., 31 (1989), pp. 40–91.

[5] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1467–1488.

[6] M. A. Celia, E. T. Bouloutas, and R. L. Zarba, *A general mass-conservative numerical solution for the unsaturated flow problem*, Water Resources Research, 7 (1990), pp. 1483–1496.

[7] F. Chaitin-Chatelin and V. Frayssé, *Lectures on Finite Precision Computation*, no. 1 in Software, Environments, and Tools, SIAM, Philadelphia, 1996.

[8] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, no. 16 in Classics in Applied Mathematics, SIAM, Philadelphia, 1996.

[9] E. Hairer, C. Lubich, and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, no. 1409 in Lecture Notes in Mathematics, Springer-Verlag, New York, 1989.

[10] W. Kampowski, P. Rentrop, and W. Schmidt, *Classification and Numerical Simulation of Electric Circuits*, Math. Inst. Tech. Univ. München, München, 1991.

[11] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[12] C. S. Kenney and A. J. Laub, *Small-scale statistical condition estimates for general matrix functions*, SIAM J. Sci. Comput., 15 (1994), pp. 36–61.

[13] W. Lioen, J. de Swart, and W. van der Veen, *Test set for IVP solvers*, tech. rep., Centrum voor Wiskunde en Informatica, Department of Numerical Mathematics, Project Group for Parallel IVP Solvers, December 23 1996.

[14] C. D. Meyer and D. Pierce, *Steps toward an iterative rank-revealing method*, Tech. Rep. ISSTECH-95-013, Boeing Information and Support Services, November 30 1995.

[15] C. T. Miller and C. T. Kelley, *A comparison of strongly convergent solution schemes for sharp front infiltration problems*, in Computational Methods in Water Resources X, Vol. 1, A. Peters, G. Wittum, B. Herrling, U. Meissner, C. Brebbia, W. Gray, and G. Pinder, eds., Kluwer Academic Publishers, 1994, pp. 325–332.

[16] Y. Mualem, *A new model for predicting the hydraulic conductivity of unsaturated porous media*, Water Resources Research, 12 (1976), pp. 513–522.

[17] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[18] L. R. Petzold, *A description of DASSL: a differential/algebraic system solver*, in Scientific Computing, R. S. Stepleman et al., ed., North Holland, Amsterdam, 1983, pp. 65–68.

[19] P. Rentrop, M. Roche, and G. Steinbach, *The applications of Rosenbrock-Wanner type methods with stepsize control in differential-algebraic equations*, Numer. Math., 55 (1989), pp. 545–563.

[20] L. A. Richards, *Capillary conduction of liquids through porous media*, Physics, 1 (1931), pp. 318–333.

[21] R. Schriber and H. B. Keller, *Driven cavity flows by efficient numerical techniques*, J. Comp. Phys., 49 (1983), pp. 310–333.

[22] L. F. Shampine, *Implementation of implicit formulas for the solution of ODEs*, SIAM J. Sci. Statist. Comput., 1 (1980), pp. 103–118.

[23] M. D. Tocci, C. T. Kelley, and C. T. Miller, *Accurate and economical solution of the pressure head form of Richards' equation by the method of lines*, Advances in Water Resources,

20 (1997), pp. 1–14.
[24] M. T. van Genuchten, *Predicting the hydraulic conductivity of unsaturated soils*, Soil Science Society of America Journal, 44 (1980), pp. 892–898.